

Estamos en esta tarea frente a un escenario APT en el que vas a asumir la personalidad de Alice Bluebird, la analista que ha sido contratada recientemente para proteger y defender a Wayne Enterprises contra varias formas de ciberataque.

En este **escenario (APT)**, los informes de la siguiente gráfica provienen de su comunidad de usuarios cuando visita el sitio web de Wayne Enterprises, y algunos de los informes referencia a los informes "P01s0n1vy". A modo de aclaración, P01s0n1vy es un grupo de

APT que ha apuntado a Wayne Enterprises. Se tiene como objetivo, como Alice, investigar el defacement con centrándonos en la reconstrucción del ataque a través de la Cadena Lockheed Martin Kill. En cuanto se vaya avanzando en la tarea veremos que vamos a trabajar el análisis de un ataque de fuerza bruta.

En todas la preguntas es necesario documentar, no solo con imágenes todas las decisiones tomadas hasta llegar a la respuesta (incluso aquellos casos en los que no se haya llegado a nada y se haya realizado un cambio de rumbo).

Para la realización de la tarea podréis acceder a material de apoyo en lo referente al uso de Splunk (apuntes, tutoriales, consultas de compañeros) pero no al uso directo para obtener respuestas de IA o de blogs de Splunk.

1. ¿Cómo ha evolucionado la cantidad de peticiones al servidor web en el tiempo, por IP origen? (Muestra el gráfico temporal tomando intervalos cada minuto entre las horas del 11 de agosto 00:00 y 00:25).

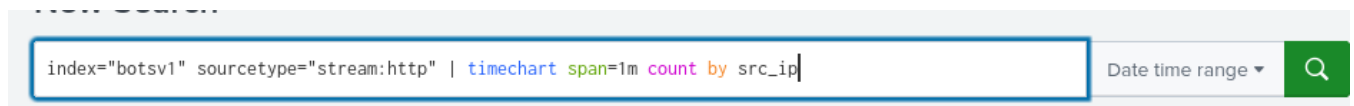
Filtra los eventos que están dentro del índice llamado botsv1, que es donde se encuentran los logs del entorno BOTS

Filtra los eventos para que solo muestre los que vienen del tipo de fuente HTTP (tráfico web monitorizado por Stream)

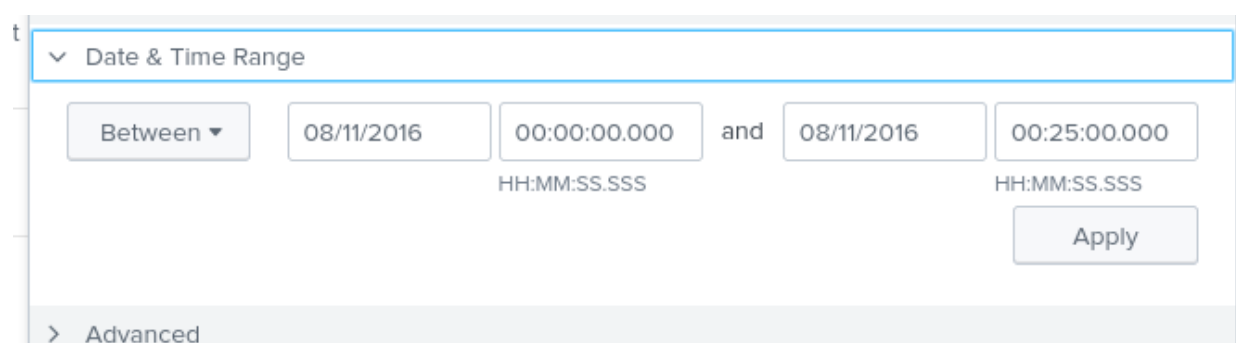
Con timechart creamos la gráfica temporal.

Con el span=1m agrupamos los eventos por intervalos de 1 minuto

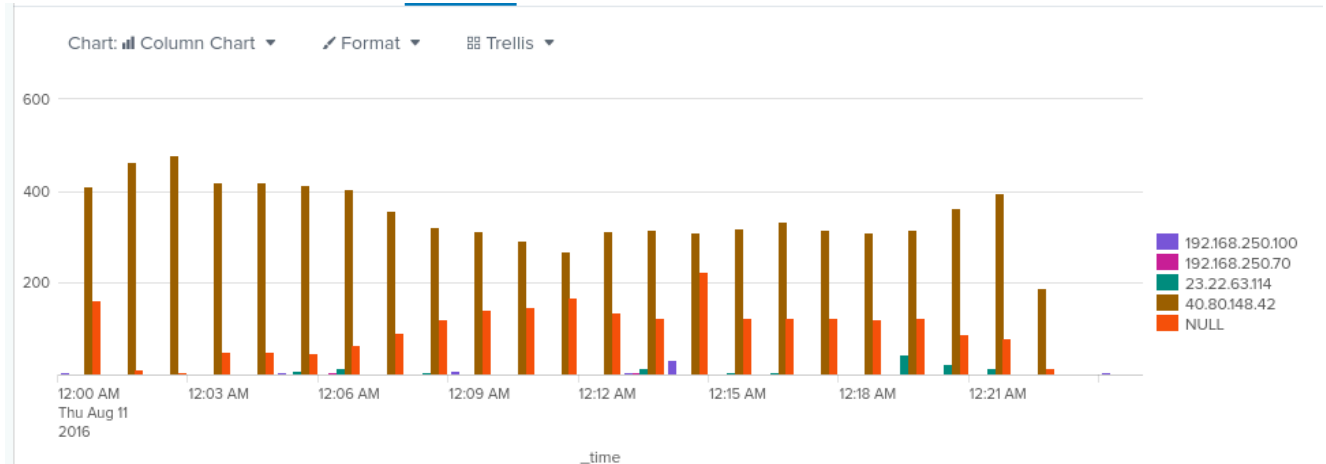
Con el scr_ip cuenta cuántos eventos HTTP hizo cada IP de origen



Ahora modificamos el date time range para poner el intervalo:



Una vez hecho eso, tenemos la gráfica solución:

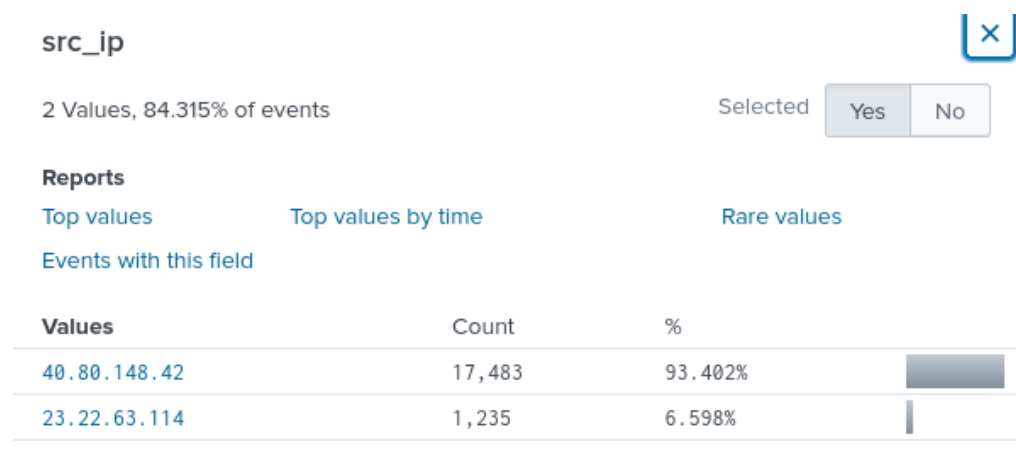


2. ¿Cuál es la dirección IP probable de alguien del grupo Po1s0n1vy que está escaneando imreallynotbatman.com en busca de vulnerabilidades en aplicaciones web?

Buscamos todos los eventos HTTP (stream:http) que contienen la cadena "imreallynotbatman.com" en cualquier campo.

```
index="botstv1" sourcetype="stream:http" "imreallynotbatman.com"
```

Ahora buscamos el campo de IP origen:



Como podemos ver, la IP 40.80.148.42 tiene una gran cantidad de solicitudes parece ser atacante.

Conclusión, esta IP hizo un número anormalmente alto de solicitudes HTTP al dominio imreallynotbatman.com, lo cual indica una actividad automatizada compatible con escaneo web, posiblemente atribuido al grupo Po1s0n1vy según el contexto del entorno BOTS.

3. ¿Cuál es la dirección IP del servidor web al que se está atacando?

Partiendo de la pregunta anterior, nos centramos solo en la IP que previamente identificamos como atacante (del grupo Po1s0n1vy)

Y calculamos las direcciones IP de destino más frecuentes en esos eventos

```
index="botsv1" sourcetype="stream:http" "imreallynotbatman.com" src_ip="40.80.148.42" | top limit=20 dest_ip
```

Vemos os resultados:

192.168.250.70	17482	99.994280
192.168.250.40	1	0.005720

La primera IP de la lista, 192.168.250.70 es la IP que buscamos.

4. ¿Qué empresa creó el escáner de vulnerabilidades web utilizado por Po1s0n1vy? Escribe el nombre de la empresa. (Por ejemplo, "Microsoft" o "Oracle")

Siguiendo los ejercicios anteriores, filtramos por la ip origen

Buscamos términos, en este caso, Microsoft.

```
index="botsv1" sourcetype="stream:http" src_ip="40.80.148.42" microsoft
```

Dentro buscamos información e encontramos:

```
src_content: areas%5b%5d=%bf'%bf%22&ordering=alpha&searchphrase=all&searchword=&task=search
src_headers: POST /joomla/index.php/component/search/ HTTP/1.1
Content-Length: 78
Content-Type: application/x-www-form-urlencoded
Referer: http://imreallynotbatman.com:80/
Cookie: ae72c62a4936b238523950a4f26f67d0=v7ikb3m59romokqmbiet3vphv3
Host: imreallynotbatman.com
Connection: Keep-alive
Accept-Encoding: gzip,deflate
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.2
Acunetix-Product: WVS/10.0 Acunetix Web Vulnerability Scanner - Free Edition)
Acunetix-Scanning-agreement: Third Party Scanning PROHIBITED
Acunetix-User-agreement: http://www.acunetix.com/wvs/disc.htm
Accept: */*

src_ip: 40.80.148.42
src_mac: 08:5B:0E:93:92:AF
src_port: 49465
status: 303
time_taken: 1070126
```

Conclusión, o que realiza o escaneo é Acunetix Web Vulnerability Scanner, se buscamos pola web vemos que pertenece a Invicti

Acunetix es una parte de Invicti Security,

5. ¿Qué sistema de gestión de contenidos es probable que esté utilizando imreallynotbatman.com?

Seguimos utilizando los filtros anteriores sin ninguna novedad

```
index="botsv1" sourcetype="stream:http" "imreallynotbatman.com"
```

Bajamos y comenzamos a buscar el campo request.

```
request: POST /joomla/index.php/component/search/ HTTP/1.1
```

Revisamos también los src_headers:

```
src_headers: POST /joomla/index.php/component/search/ HTTP/1.1
```

Tras esto, vemos que hace referencia al gestor de contenidos JOOMLA que es el que probablemente esté utilizando.

6. ¿Cuál es el nombre del archivo que desfiguró el sitio web imreallynotbatman.com? Solo el nombre del archivo con la extensión (por ejemplo, "notepad.exe" o "favicon.ico"). Nota: analiza el tráfico de red en el sentido inverso (de nuestro servidor web hacia una ip externa del atacante).

Siguiendo la línea anterior, buscamos siguiendo la ip origen del servidor:

```
index="botsv1" sourcetype="stream:http" src_ip="192.168.250.70"
```

Vamos a uri_path e inspeccionamos:

uri_path

4 Values, 100% of events

Selected

Yes

No

Reports

Top values

Top values by time

Rare values

Events with this field

Values	Count	%	
/core/list.xml	2	28.571%	
/jed/list.xml	2	28.571%	
/poisonivy-is-coming-for-you-batman.jpeg	2	28.571%	
/core/extensions/com_joomlaupdate.xml	1	14.286%	

Aquí podemos ver el archivo entregado por el servidor que pudo desconfigurar el sitio web.

Además hace referencia a poisonivy que es el grupo sospechoso:

/poisonivy-is-coming-for-you-batman.jpeg	2	28.571%
--	---	---------

7. Este ataque utilizó DNS dinámico para resolver la IP maliciosa. ¿Cuál es el nombre de dominio completo (FQDN) asociado con este ataque?

Accedemos al fichero jpeg malicioso para filtrar por este paquete que fue el que desconfiguró la red.

También contamos cuántas veces se produjo esa petición, agrupada por el campo site.

```
index="botsv1" sourcetype="stream:http" src_ip="192.168.250.70" | spath request | search request="GET /poisonivy-is-coming-for-you-batman.jpeg HTTP/1.0" | chart count by site
```

Ahora una tabla dice cuántas veces se solicitó el archivo poisonivy-is-coming-for-you-batman.jpeg y desde qué site se hizo.

prankglassinebracket.jumpingcrab.com:1337	2
---	---

Comprobamos que realmente solicitó el archivo:

```
index="botsv1" sourcetype="stream:http" src_ip="192.168.250.70" | spath request | search request="GET /poisonivy-is-coming-for-you-batman.jpeg HTTP/1.0" site="prankglassinebracket.jumpingcrab.com:1337"
```

Buscamos:

```
request: GET /poisonivy-is-coming-for-you-batman.jpeg HTTP/1.0
```

Conclusión, el nombre de dominio completo es:

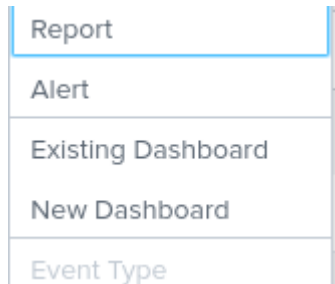
prankglassinebracket.jumpingcrab.com:1337

8. Crea un dashboard en el que se vea en forma de gráfico de secciones las ip que consultan nuestro servidor web. (Se puede obtener haciendo una consulta en una web externa)

Creamos una tabla que cuenta el número de eventos (peticiones HTTP) para cada dirección IP de origen (src_ip).

```
index="botsv1" sourcetype="stream:http" dest_ip="192.168.250.70" | chart count by src_ip
```

Ahora vamos a crear y accedemos a New Dashboard:



Una vez dentro:

Save Panel to New Dashboard

×

Dashboard Title

Ejercicio 8

ejercicio_8 Edit ID

Description

Optional

Permissions

Private

How do you want to build your dashboard?

[What's this?](#)

Classic Dashboards

The traditional Splunk dashboard builder

Dashboard Studio NEW

A new builder to create visually-rich, customizable dashboards

Panel Title

Ejercicio 8

Visualization Type

☒ Statistics Table

> Advanced Panel Settings

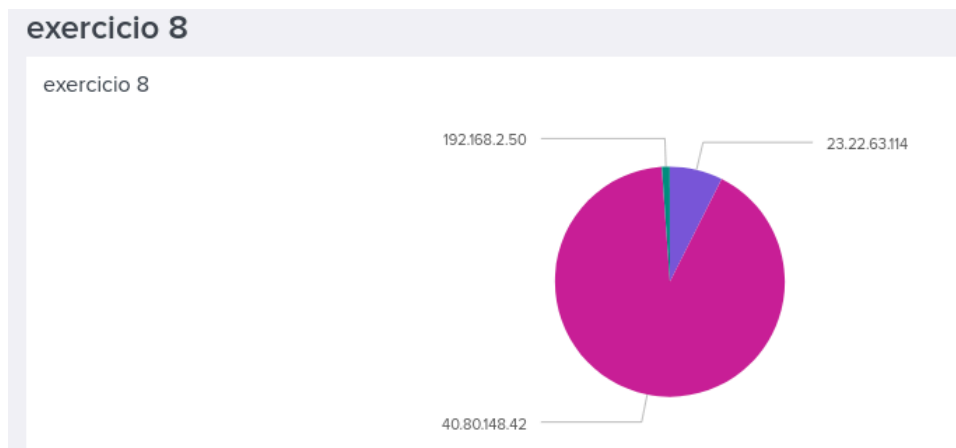
Cancel

Save to Dashboard

Cancel

Save to Dashboard

Lo creamos y vemos:



Aquí tenemos el dashboard, en el cual se ven las ip que consultan nuestro servidor web

9. ¿Qué dirección IP ha vinculado Po1s0n1vy a los dominios que están preconfigurados para atacar a Wayne Enterprises?

Usando el ejercicio 6, utilizamos lo mismo:

```
index="botsv1" | sourcetype="stream:http" src_ip="192.168.250.70"
```

Buscamos e encontramos:

```
data_packets_in: 2
data_packets_out: 0
dest_ip: 23.22.63.114
dest_mac: 08:5B:0E:93:92:AF
dest_port: 1337
duplicate_packets_in: 2
duplicate_packets_out: 0
endtime: 2016-08-10T22:13:46.915172Z
http_method: GET
missing_packets_in: 0
missing_packets_out: 0
network_interface: eth1
packets_in: 6
packets_out: 5
reply_time: 0
request: GET /poisonivy-is-coming-for-you-batman.jpeg HTTP/1.0
request_ack_time: 3246
request_time: 61714
response_ack_time: 0
response_time: 0
server_rtt: 32357
server_rtt_packets: 2
server_rtt_sum: 64714
site: prankglassinebracket.jumpingcrab.com:1337
src_headers: GET /poisonivy-is-coming-for-you-batman.jpeg HTTP/1.0
Host: prankglassinebracket.jumpingcrab.com:1337
```

Si buscamos el paquete en el que se envió el fichero malicioso, vemos que está asociado al dominio que tiene relacionada la IP **23.22.63.114**

10. Según los datos recopilados de este ataque y fuentes comunes de inteligencia de código abierto para nombres de dominio, ¿cuál es la dirección de correo electrónico más probable asociada con el grupo APT Po1s0n1vy?

Recordando el dominio encontrado anteriormente, prankglassinebracket.jumpingcrab.com y también el grupo APT Po1s0n1vy


Nos centramos más en el grupo Po1s0n1vy
Buscamos y encontramos:

 Open Threat Exchange
https://otx.alienvault.com › po1s0... · Traducir esta página

Domain: po1s0n1vy.com - LevelBlue - Open Threat Exchange

Analysis Overview IP Address Domain Not Currently Resolving to an IP WHOIS Registrar: TUCOWS, INC., Creation Date: Jul 21, 2016

Y también:

 Open Threat Exchange
https://otx.alienvault.com › LILLI... · Traducir esta página

Email: LILLIAN.ROSE@PO1S0N1VY.COM - AlienVault OTX

Learn about the latest cyber threats. Research, collaborate, and share threat intelligence in real time. Protect yourself and the community against today's ...

Suponemos que el correo asociado al grupo es:
LILLIAN.ROSE@PO1S0N1VY.COM

11. ¿Qué URIs parecen relacionadas con autenticación?

12. ¿Qué dirección IP probablemente está intentando un ataque de fuerza bruta contra imreallynotbatman.com?

Buscamos eventos donde la IP de destino es el servidor web atacado.

Nos centramos en peticiones HTTP POST, que son las que suelen usarse para enviar datos a un servidor.

Agrupamos los eventos por IP de origen (src_ip) y por el contenido del formulario enviado (form_data).

```
index="botsv1" sourcetype=stream:http dest_ip=192.168.250.70 http_method=POST | stats count by src_ip, form_data
```

Este comando permite identificar qué datos están enviando los atacantes al servidor mediante peticiones POST, y desde qué IP lo están haciendo

Events	Patterns	Statistics (9,024)	Visualization
Show: 20 Per Page ▾		Format ▾	Preview: On
		< Prev	1 2 3 4 5 6 7 8
src_ip	form_data		
23.22.63.114	username=admin&0960d493674eb04861bd64da9b662118=1&task=login&return=aW5kZXgucGhw&option=com_login&passwd=arthur		
23.22.63.114	username=admin&0edae02d7478dfb41641700ef384807a=1&task=login&return=aW5kZXgucGhw&option=com_login&passwd=bigdaddy		
23.22.63.114	username=admin&115c3aa6072f4b02b4354909431510f6=1&task=login&return=aW5kZXgucGhw&option=com_login&passwd=blazer		
23.22.63.114	username=admin&12c709bcc2e14d5a015f054d18d36537=1&task=login&return=aW5kZXgucGhw&option=com_login&passwd=fire		

13. ¿Cuál es el nombre del archivo ejecutable subido por Po1s0n1vy? Por favor, incluye la extensión del archivo. (Por ejemplo, "notepad.exe" o "favicon.ico")

multipart/form-data (usado en formularios web para subir archivos), e busco que en algún campo aparezca una cadena que termine en .exe.

Buscamos .exe o parecidos:

[illegible]

3791.exe → Es el archivo subido por Po1s0n1vy

14. ¿Cuál fue la primera contraseña de fuerza bruta utilizada? (Puedes investigar sobre el comando transaction)

Siguiendo la línea y poniendo la ip origen del equipo atacante.

Además stats count by form_data, _time: cuenta cuántas veces aparece cada valor del campo form_data (contenido enviado por POST) en cada momento (_time)

```
index="botsv1" sourcetype=stream:http dest_ip=192.168.250.70 http_method=POST src_ip=23.22.63.114 | stats count by form_data, _time
```

index="botsv1" sourcetype=stream:http dest_ip=192.168.250.70 http_method=POST
src_ip=23.22.63.114 | stats count by form_data, _time

Con esto obtenemos la primera contraseña utilizada(primer fecha):

Events Patterns Statistics (412) Visualization		
Show: 20 Per Page	Format	Preview: On
<div> <div>< Prev</div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>...</div> <div>Next ></div> </div>		
form_data	_time	count
username=admin&task=login&return=aW5kZXgucGhw&option=com_login&passwd=12345678&9d873c2becd118318849d13cf18b60ff=1	2016-08-10 23:45:21.226	1
username=admin&863349a657c211fbfeb90ebe9427654c=1&task=login&return=aW5kZXgucGhw&option=com_login&passwd=letmein	2016-08-10 23:45:21.241	1
username=admin&task=login&return=aW5kZXgucGhw&option=com_login&passwd=qwerty&af4df60674155567dee0566f87045251=1	2016-08-10 23:45:21.247	1
username=admin&task=login&return=aW5kZXgucGhw&option=com_login&passwd=1234&aaf6297ae5c1e3df78a421bc55548d16=1	2016-08-10 23:45:21.250	1
username=admin&task=login&return=aW5kZXgucGhw&option=com_login&76e93e8488d9a46878468d88954a0d54=1&passwd=123456	2016-08-10 23:45:21.260	1

15. ¿Cuál fue la contraseña correcta para acceder como administrador al sistema de gestión de contenidos que ejecuta "imreallynotbatman.com"?

Siguiendo la misma línea, ...

```
index="botsv1" http_method=POST dest_ip=192.168.250.70 sourcetype="stream:http"
| rex field=form_data "passwd=(?<Password>\w+)"
| stats count values(src) by Password
| sort - count
```

Así realizamos una búsqueda de peticiones POST hacia la IP 192.168.250.70, extrayendo las contraseñas enviadas en el campo form_data de esas solicitudes. Luego, cuenta cuántas veces aparece cada contraseña, muestra las direcciones IP de origen asociadas con esas contraseñas y las organiza en orden descendente según la frecuencia con la que se usaron.

Events	Patterns	Statistics (412)	Visualization
Show: 20 Per Page ▾	Format ▾	Preview: On	< Prev 1 2 3 4 5 6
Password ▾	count ▾	values(src) ▾	
batman	2		
000000	1		
1111	1		
111111	1		
11111111	1		
112233	1		
1212	1		

Como podemos ver, la password que sale dos veces es batman. Por lo que es la contraseña a utilizar.

Ahora, seguimos...

Dentro de las peticiones POST, se busca específicamente aquellas que acceden a la URL /joomla/Administrator/index.php, que es típicamente la ruta del panel de administración de Joomla. También se añade el filtro status=200 para quedarse solo con las solicitudes que fueron aceptadas por el servidor, es decir, accesos que tuvieron éxito (lo que indica que el login fue correcto).

Después de filtrar los eventos, el comando stats count by src_ip cuenta cuántas veces cada dirección IP realizó ese tipo de solicitud, lo que ayuda a identificar qué IP logró acceder con éxito al sistema. Finalmente, se utiliza el comando rex para examinar el contenido del campo form_data, que contiene los datos enviados en los formularios. Este rex extrae el nombre de usuario y la contraseña utilizadas, buscando coincidencias exactas con username=admin y passwd=batman

```
index="botsv1" sourcetype=stream:http dest_ip=192.168.250.70 http_method=POST
uri=/joomla/Administrator/index.php status=200 | stats count by src_ip | rex
field=form_data "username=(?<username>admin).*passwd=(?<password>batman)"
```

```
index="botsv1" sourcetype=stream:http dest_ip=192.168.250.70 http_method=POST uri=/joomla/Administrator/index
.php status=200 | stats count by src_ip | rex field=form_data "username=(?<username>admin).*passwd
=(?<password>batman)"
```

Nos aparece:

Events	Patterns	Statistics (1)	Visualization
Show: 20 Per Page ▾	Format ▾	Preview: On	
src_ip ▾	count ▾		
40.80.148.42	12		

Con esto sacamos la conclusión de que se realizaron 12 accesos exitosos. Lo que confirmamos que la contraseña es: **batman**

16. ¿Cuál fue la longitud promedio de las contraseñas utilizadas en el intento de fuerza bruta? (Redondea al número entero más cercano. Por ejemplo, "5" y no "5.23213")

Siguiendo la línea de la pregunta anterior:

Este comando se centra en analizar las contraseñas que se están enviando a la ruta /joomla/Administrator/index.php, es decir, el panel de administración de Joomla, mediante peticiones HTTP de tipo POST.

Primero, con `rex field=form_data "passwd=(?<password>\w+)"`, se extrae la contraseña del campo `form_data` que llega en la solicitud. Este campo suele contener los datos del formulario de inicio de sesión, por lo tanto se busca el valor que sigue a `passwd=` y se guarda en un nuevo campo llamado `password`.

Luego, `eval length = len(password)` calcula la longitud de cada contraseña extraída y la almacena en un nuevo campo llamado `length`.

A continuación, `stats avg(length) as avglength` calcula la longitud media de todas las contraseñas extraídas y la guarda bajo el nombre `avglength`.

Finalmente, con `eval rounded = round(avglength,0)`, se redondea ese valor medio al número entero más cercano, almacenándolo en un nuevo campo llamado `rounded`.

```
index="botsv1" sourcetype=stream:http dest_ip=192.168.250.70 http_method=POST uri=/joomla/Administrator/index.php | rex field=form_data "passwd=(?<password>\w+)" | eval length = len(password) | stats avg(length) as avglength | eval rounded = round(avglength,0)
```

Nos aparece:

Show: 20 Per Page ▾	Format ▾	Preview: On
avglength ▾		rounded ▾
6.174334140435835		6

Por tanto, la longitud promedio es **6**.

17. ¿Cuántos segundos transcurrieron entre el escaneo de fuerza bruta que identificó la contraseña correcta y el inicio de sesión comprometido? Redondea a 2 decimales.

Siguiendo la línea del anterior...

Analizamos los intentos de autenticación al sistema Joomla, concretamente las solicitudes POST dirigidas al archivo /joomla/Administrator/index.php, que es donde normalmente se realizan los inicios de sesión administrativos en Joomla. Es decir, se está inspeccionando el tráfico relacionado con posibles logins.

A continuación, extraemos la contraseña enviada en el formulario mediante una expresión regular. El campo `form_data` contiene los datos enviados a través del POST, y con el uso de `rex`, se busca el valor que corresponde a `passwd=`. Ese valor se guarda en un nuevo campo llamado `password`, permitiendo trabajar con él directamente.

Después, se realizó un filtrado para quedarme solo con los eventos en los que se utilizó exactamente la contraseña "batman". Esto nos permite centrarnos en los intentos

específicos donde se usó esa clave para intentar acceder como administrador al sitio.

Una vez filtrados esos eventos, se agrupan como una transacción usando el campo password. Esto sirve para unir todos los eventos relacionados que forman parte del mismo intento o sesión con esa contraseña, y así poder calcular cuánto tiempo duró esa secuencia de acciones.

Finalmente, se extrae la duración de dicha transacción en una tabla, y luego se redondea ese valor usando eval para obtener una cifra entera. Este número representa, en segundos, cuánto tiempo duró el intento de acceso usando la contraseña "batman".

```
index="botsv1" sourcetype=stream:http dest_ip=192.168.250.70 http_method=POST uri=/joomla/Administrator/index.php | rex field=form_data "passwd=(?<password>\w+)" | search password="batman" | transaction password | table duration | eval rounded_duration = round(duration,0)
```

Nos sale:

Show: 20 Per Page ▾ Format ▾ Preview: On	
duration ▾	rounded_duration ▾
92.169084	92

Al redondear el valor, la duración es **92.17 segundos**.

18. ¿Cuántas contraseñas únicas se intentaron en el ataque de fuerza bruta?

Siguiendo en la misma línea del trabajo ...

Busqué analizar las contraseñas enviadas en intentos de autenticación al panel de administración de Joomla, específicamente mediante solicitudes POST dirigidas a la URL /joomla/Administrator/index.php.

Primero, el comando rex se usa para extraer del campo form_data la contraseña enviada por el usuario. Mediante una expresión regular, se identifica el valor que aparece después de passwd= y se almacena en un nuevo campo llamado password. De este modo, ahora puedo trabajar directamente con las contraseñas usadas.

Después, el comando dedup password elimina las duplicaciones. Es decir, si una misma contraseña fue usada varias veces, solo se considerará una vez. Esto sirve para saber cuántas contraseñas distintas han sido probadas en los intentos de login.

Por último, con stats count, se cuenta cuántas contraseñas únicas se han encontrado. El resultado final te da un número que representa la cantidad de contraseñas diferentes que fueron utilizadas en los intentos de inicio de sesión al administrador Joomla.

```
index="botsv1" sourcetype=stream:http dest_ip=192.168.250.70 http_method=POST uri=/joomla/Administrator/index.php | rex field=form_data "passwd=(?<password>\w+)" | dedup password | stats count
```

Me sale:

Events	Patterns	Statistics (1)	Visualization
Show: 20 Per Page ▾ / Format ▾ <input checked="" type="checkbox"/> Preview: On			
count ↕			
412			

Por lo que he obtenido un total de **412 contraseñas únicas**.

19. Escribe una regla sigma que compruebe si el Windows Security Event ID (EventID) es 4656 o 4663. (Es suficiente con poner solo los campos obligatorios).

La regla sigma es la siguiente:

```
title: Detect Access or Audit Events (4656 or 4663)
logsource:
  product: windows
  service: security
detection:
  selection:
    EventID:
      - 4656
      - 4663
  condition: selection
```

Esta regla Sigma que he creado está diseñada para detectar eventos del registro de seguridad de Windows que correspondan a los identificadores de evento 4656 o 4663.

El Event ID 4656 se refiere a intentos de acceso a objetos en el sistema (como archivos o claves de registro), mientras que el 4663 indica un acceso efectivo a dichos objetos.

La sección logsource especifica que esta regla debe aplicarse a registros de tipo "security" en sistemas Windows.

En el bloque detection, se define una condición llamada selection que filtra eventos cuyo campo EventID sea 4656 o 4663. Finalmente, la condición selection se activa si se cumple alguno de esos valores

20. Dada la siguiente consulta en Splunk **escribe la regla sigma** equivalente

```
source="WinEventLog:*" AND ((Image="*\\excel.exe" OR Image="*\\mspub.exe" OR  
Image="*\\onenote.exe" OR Image="*\\onenoteim.exe" OR Image="*\\outlook.exe"  
OR Image="*\\powerpnt.exe" OR Image="*\\winword.exe") AND  
ImageLoaded="*\\kerberos.dll")
```

La regla que creé es:

```
title: Office Application Loading kerberos.dll  
description: Detecta la carga de kerberos.dll  
author: Martin Losada  
logsource:  
  product: windows  
  category: process_load  
detection:  
  selection:  
    Image|endswith:  
      - '\\excel.exe'  
      - '\\mspub.exe'  
      - '\\onenote.exe'  
      - '\\onenoteim.exe'  
      - '\\outlook.exe'  
      - '\\powerpnt.exe'  
      - '\\winword.exe'  
    ImageLoaded|contains: 'kerberos.dll'  
  condition: selection
```

Esta regla Sigma que he creado está pensada para detectar eventos en sistemas Windows relacionados con la carga de bibliotecas dinámicas (DLLs) por parte de aplicaciones de Office. Me enfoco en eventos de tipo process_load, que registran las DLLs que cada proceso carga en tiempo de ejecución.

Utilizo el operador endswith para identificar los ejecutables de Office y contains para comprobar si se ha cargado el archivo kerberos.dll. Este comportamiento puede ser un indicador de ataques avanzados que utilizan Office como vector inicial de ejecución y manipulan bibliotecas sensibles como la de Kerberos para obtener privilegios o persistencia.